Automatic Speech Recognition (CS753)

Lecture 22: Speaker Adaptation & Pronunciation modelling

Instructor: Preethi Jyothi

Apr 10, 2017

Speaker variations

- Major cause of variability in speech is the differences between speakers
    - Speaking styles, accents, gender, physiological differences, etc.
- Speaker independent (SI) systems: Treat speech from all different speakers as though it came from one and train acoustic models
- Speaker dependent (SD) systems: Train models on data from a single speaker
- Speaker adaptation (SA): Start with an SI system and adapt using a small amount of SD training data

Types of speaker adaptation

- **Batch/Incremental adaptation**: User supplies adaptation speech beforehand *vs.* system makes use of speech collected as the user uses a system

- **Supervised/Unsupervised adaptation**: Knowing transcriptions for the adaptation speech *vs.* not knowing them

- **Training/Normalization**: Modify only parameters of the models observed in the adaptation speech *vs.* find transformation for all models to reduce cross-speaker variation

- **Feature/Model transformation**: Modify the input feature vectors *vs.* modifying the model parameters.

Normalization

- Cepstral mean and variance normalization: Effectively reduce variations due to channel distortions

$$\mu_f = \frac{1}{T}\sum_t f_t$$

$$\sigma_f{}^2 = \frac{1}{T}\sum_t (f_t^2 - \mu_{f,t}^2)$$

$$\hat{f}_t = \frac{f_t - \mu_f}{\sigma_f}$$

- Mean subtracted from the cepstral features to nullify the channel characteristics

Speaker adaptation

- Speaker adaptation techniques can be grouped into two families:

    1. Maximum a posterior (MAP) adaptation

    2. Linear transform-based adaptation

Speaker adaptation

- Speaker adaptation techniques can be grouped into two families:

1. Maximum a posterior (MAP) adaptation

2. Linear transform-based adaptation

Maximum a posterior adaptation

- Let λ characterise the parameters of an HMM and Pr(λ) be prior knowledge. For observed data X, the maximum a posterior (MAP) estimate is defined as:

$$\begin{aligned}\lambda^* &= \arg\max_{\lambda} Pr(\lambda|X) \\ &= \arg\max_{\lambda} Pr(X|\lambda) \cdot Pr(\lambda)\end{aligned}$$

- If Pr(λ) is uniform, then MAP estimate is the same as the maximum likelihood (ML) estimate

Recall: ML estimation of GMM parameters

ML estimate:

$$\mu_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m) x_t}{\sum_{t=1}^{T} \gamma_t(j,m)}$$

- where $\gamma_t(j, m)$ is the probability of occupying mixture component m of state j at time t

MAP estimation

ML estimate:

$$\mu_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m) x_t}{\sum_{t=1}^{T} \gamma_t(j,m)}$$

- where $\gamma_t(j, m)$ is the probability of occupying mixture component m of state j at time t

MAP estimate:

$$\hat{\mu}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m)}{\tau + \sum_{t=1}^{T} \gamma_t(j,m)} \bar{\mu}_{jm} + \frac{\tau}{\tau + \sum_{t=1}^{T} \gamma_t(j,m)} \mu_{jm}$$

- where $\bar{\mu}_{jm}$ is ML estimate of the mean of the adaptation data, μ_{jm} is prior mean chosen from previous EM iteration, τ controls the bias between prior and information from the adaptation data

MAP estimation

- MAP estimate is derived after 1) choosing a specific prior distribution for $\lambda = (c_1,\ldots,c_m, \mu_1,\ldots,\mu_m, \Sigma_1,\ldots,\Sigma_m)$ 2) updating model parameters using EM

- Property of MAP: Asymptotically converges to ML estimate as the amount of adaptation data increases

- Updates only those parameters which are observed in the adaptation data

Speaker adaptation

- Speaker adaptation techniques can be grouped into two families:

1. Maximum a posterior (MAP) adaptation

2. Linear transform-based adaptation

Linear transform-based adaptation

- Estimate a linear transform from the adaptation data to modify HMM parameters

- Estimate transformations for each HMM parameter? Would require very large amounts of training data.
    - Tie several HMM states and estimate one transform for all tied parameters
    - Could also estimate a single transform for all the model parameters

- Main approach: Maximum Likelihood Linear Regression (MLLR)

MLLR

- In MLLR, the mean of the m-th Gaussian mixture component μ_m is adapted in the following form:

$$\hat{\mu}_m = A\mu_m + b_m = W\xi_m$$

where $\hat{\mu}_m$ is the adapted mean, $W = [A, b]$ is the linear transform and ξ_m is the extended mean vector, $[\mu_m^T, 1]^T$

- W is estimated by maximising the likelihood of the adaptation data X:

$$W^* = \arg\max_W \{\log \Pr(X; \lambda, W)\}$$

- EM algorithm is used to derive this ML estimate

Regression classes

- So far, assumed that all Gaussian components are tied to a global transform
- Untie the global transform: Cluster Gaussian components into groups and each group is associated with a different transform
- E.g. group the components based on phonetic knowledge
  - Broad phone classes: silence, vowels, nasals, stops, etc.
  - Could build a decision tree to determine clusters of components

Lexicons and Pronunciation Models

Pronunciation Dictionary/Lexicon

- Link between phone-based HMMs in the acoustic model and words in the language model

- Derived from language experts: Sequence of phones written down for each word

- Dictionary construction involves:

    1. Selecting what words to include in the dictionary

    2. Pronunciation of each word (also, check for multiple pronunciations)

Graphemes vs. Phonemes

- Instead of a pronunciation dictionary, could represent a pronunciation as a sequence of graphemes (or letters)

- Main advantages:

1. Avoid the need for phone-based pronunciations

2. Avoid the need for a phone alphabet

3. Works pretty well for languages with a direct link between graphemes (letters) and phonemes (sounds)

Grapheme-based ASR

Language	ID	System	WER (%)		
			Vit	CN	CNC
Kurmanji Kurdish	205	Phonetic	67.6	65.8	64.1
		Graphemic	67.0	65.3	
Tok Pisin	207	Phonetic	41.8	40.6	39.4
		Graphemic	42.1	41.1	
Cebuano	301	Phonetic	55.5	54.0	52.6
		Graphemic	55.5	54.2	
Kazakh	302	Phonetic	54.9	53.5	51.5
		Graphemic	54.0	52.7	
Telugu	303	Phonetic	70.6	69.1	67.5
		Graphemic	70.9	69.5	
Lithuanian	304	Phonetic	51.5	50.2	48.3
		Graphemic	50.9	49.5	

Image from: Gales et al., Unicode-based graphemic systems for limited resourcee languages, ICASSP 15

Graphemes vs. Phonemes

- Instead of a pronunciation dictionary, could represent a pronunciation as a sequence of graphemes (or letters)

- Main advantages:

1. Avoid the need for phone-based pronunciations

2. Avoid the need for a phone alphabet

3. Works pretty well for languages with a direct link between graphemes (letters) and phonemes (sounds)

Grapheme to phoneme (G2P) conversion

- Produce a pronunciation (phoneme sequence) given a written word (grapheme sequence)
- Useful for:
  - ASR systems in languages with no pre-built lexicons
  - Speech synthesis systems
  - Deriving pronunciations for out-of-vocabulary (OOV) words

G2P conversion (I)

- One popular paradigm: Joint sequence models [BN12]
  - Grapheme and phoneme sequences are first aligned using EM-based algorithm
  - Results in a sequence of graphones (joint G-P tokens)
  - Ngram models trained on these graphone sequences
- WFST-based implementation of such a joint graphone model [Phonetisaurus]

[BN12]:Bisani & Ney , “Joint sequence models for grapheme-to-phoneme conversion”,Specom 2012

[Phonetisaurus] J. Novak, Phonetisaurus Toolkit

G2P conversion (II)

- Neural network based methods are the new state-of-the-art for G2P

- Bidirectional LSTM-based networks using a CTC output layer [Rao15]. Comparable to Ngram models.

- Incorporate alignment information [Yao15]. Beats Ngram models.

[Rao15] Grapheme-to-phoneme conversion using LSTM RNNs, ICASSP 2015
[Yao15] Sequence-to-sequence neural net models for G2P conversion, Interspeech 2015